Universe Scientific Publishing

ORIGINAL ARTICLE

Application and Feasibility Analysis of Proactive Anti-cheating Algorithm Based on ML

Ruixi Liu*

Zhengzhou Linghang Experimental School, Zhengzhou 45000, China. E-mail: handsomeroy@hotmail.com

Abstract: Nowadays, most Internet application contents based on modern Internet service providers are realized through human-computer interaction. Therefore, the application scenarios of HCI in life are very wide, and the visual operation path provides great system availability and user-friendliness. And its security is an important fulcrum for service providers (SP) to obtain benefits, or fund transactions between terminals. According to the trigger mechanism of cheating, it is mainly divided into exposure cheating, click cheating, and conversion cheating. Cheating will greatly pollute the network environment and it even violates laws and regulations. Therefore, ISPs are deploying and continuously improving anti-cheat algorithms and mechanisms during the service development stage. The location of these operating mechanisms is generally the network server used by the SP to provide services, and some algorithms and monitoring behaviors significantly increase the load of the server, thereby increasing service costs.

Based on machine learning (ML) technology, this paper introduces a feasibility analysis of anti-cheat algorithm for human operation based on object terminal, which is suitable for behavior recognition level and data level. The algorithm learns user behavior through their active participation. It finds the path of the request by combining with packet matching of abnormal behavior characteristic of cheating behavior updated regularly by the service provider. This algorithm also detects cheating behavior through local controller. When the two peers have the same or a high probability of prejudgment after calculation, the server is triggered to actively identify user behavior, using the trust profile to analyze and log the cheating path. The compatibility and friendliness of terminal equipment characteristics and hardware level with AI algorithms in the current environment is helpful to reduce the load caused by the server mounting of the ISP and costs.

Keywords: Human-Computer Interaction (HCI); Machine Learning (ML); Artificial Intelligence (AI); Anti-cheat; Algorithm; Localization

1. Introduction

Machine learning is an important part in the field of artificial intelligence. This kind of program can promote the performance of this logic evaluation by evaluating a specific performance index of the task and learning the experience gained from its index evaluation to reach an increasingly perfect level. If this goal can be achieved, the corresponding program will be interpreted as the ability to learn from experience, which is called machine learning (ML). It takes classification, clustering, regression, and dimensionality reduction as the main principles of learning.[1]

Processor chips produced after 2015 can reach a relatively sufficient level in terms of computing frequen-

doi: 10.18686/esta.v7i2.141

-cy. Our study found that in the daily human-computer interaction scenarios, the utilization rate of the processor did not exceed 50% in 75.51% cases and 75% in 97.33% cases. Therefore, the equipment processing chips produced after that have sufficient capacity to merge processing and calculate the anti-cheating mechanism based on behavior monitoring.

In this article, an anti-cheat algorithm strategy based on ML peer-to-peer systems is proposed. Since the peer node is responsible for calculating the path, the feasibility of a single child node to judge cheating goes very high. Since the intent cheater has the ultimate terminal to implement cheating, all actions of the cheater have a high probability of running on this terminal, so the key goal is to design a low-cost path and efficient cheating behavior using the terminal as the computing carrier computer system, thus replacing the operation of the original single node anti-cheat mechanism.

2. Problem description

Path discovery requires a lot of resources, especially in data carriers with many instantaneous users, such as online games or online shopping. Therefore, an efficient and cheap method is needed, otherwise the Internet provider's anti-cheating mechanism will consume a huge amount of server memory, increasing the operating cost. In the human-computer interaction experiment, most of the user's operations are converted into digital signals from the computer's external equipment and then they are collected by it. But for the purpose of realizing a function, it needs to run an algorithm in the background to determine the path to be followed. Considering that the operation to complete a specific goal may have different logical properties, this may be a complex decision operation, and it will be more complicated in a specific irregular artificial environment. Therefore, a near-optimal path is necessary if it is suitable in terms of computation time and memory. [2]

To explain this problem accurately, a location model from the source point S to the destination point D is designed. According to the distribution model, the regional path $Z_8 \rightarrow Z_6 \rightarrow Z_5 \rightarrow Z_1$ is set. Then, in the conversion of each regional grid level, drawing a data distribution map of all normal points and discrete points, and sampling the data delivered from the "experience" learned in the above principles and the server of the service provider are what to do. The next step is to compare, and when the shortest distance between a feature and other feature values exceeds the threshold of the system, the server should be alerted in advance, and an anti-cheat mechanism is triggered in both directions. The aim of this paper is to find an automatic anti-cheating path split into two parts for separate deployment at the given source and target with the minimum computational cost.

3. Solution

Initially, the design and implementation of this problem was a simple multi-terminal "clustering anomaly detection" mechanism supplemented by data packets. But for this problem, according to the existing information, it cannot be realized normally. The main anti-cheating mechanism is composed of "green dill algorithm" represented by engine optimization and "abnormality discovery model" represented by behavior monitoring. The difficulties are as follows:

First, it needs to be ensured that the original web server running the anti-cheat algorithm and the user terminal mentioned in this design remain stable in the same time domain. To solve this problem, the terminal is regarded as a node, and a specific node is identified within a short time range in the transmission path of the data packet during the connection between the terminal and the server. Then a set of packets is mounted with localized behavior identification and interaction with the server. However, regarding the location of the target terminal, the data content between any terminal and the server is not unique and continuous. The reliability of the data packet content is limited.

Secondly, in the human-computer interaction application scenario based on application nesting environment, it is not easy to deploy a computer program with independent running ability locally, and the user's hardware performance is uneven, which makes it impossible to accurately predict the feasibility of the behavior. So, it is not easy to find a credible path.

Therefore, to overcome these limitations, some rule changes and logical adaptations are required. First, to correctly understand the proposed anti-cheat algorithm, this article briefly introduces the path discovery

algorithm through participating nodes. Detailed algorithm aspects can be found as follows. Second, to correctly understand the advantages of this strategy, this article briefly analyzes the feasibility and application mechanism of this strategy.

3.1 Apply page-rank-based weight model

Page Rank is an algorithm that sorts the importance of the triggered content. It initializes all the weights of the page to the same value when treating it as a rectangular directed graph with four endpoints. It starts from an arbitrary point in a random page and then it jumps to the link attached to it, where the web pages are regarded as nodes. Starting from page A, the probability of jumping to B, C, and D are all 1/3, so it can be deduced that if a link has K outgoing links, then the probability of the target position of the next jump is 1 /K. Similarly, the jump probability from B to C is 0, and D has a 1/2 probability to jump to B or C. Then for every link that goes out of the chain (I) there is a matrix F[i][j] = 1/K. With the appearance of the iterative jump process, the weights will also be propagated to the next level of pages. The weight value after the stable iteration occurs is called the Page Rank Value, which are going with the probability of eventually stopping pointing.

So, it can be deduced that:

$$V'=\alpha FV + (1-\alpha)e$$

At the same time, using the ranking mechanism BRank to measure the importance of the page to change the weight of the search results and using the reverse pointer (to increase the value of Page Rank) to implement anti-cheat are crucial. The local blacklist is established and maintained in the server and the BRank value E(x) is assigned to the links in the blacklist for majority assignment. Then if a link is nested with links with blacklist directivity, then BRank will be assigned repeatedly.

Through analogy, the formula is:

$$BR(x) = E(x)(1-d)+d(BR(t1)/C(t1)+\ldots BR(tn)/C(tn))$$

(C(tn) represents the number of tn chains and d represents the damping coefficient)

For this topic, it is necessary to establish a simpler whitelist mechanism. Then a whitelist mechanism like the Hilltop algorithm is introduced, so that it can not only mark users with lower frequency to avoid the risk of obtaining a larger BRank value, but also balance the data link.

3.2 Localized deployment and implementation of anti-cheating algorithm

The basic communication model combines different aspects of client-server and peer-to-peer communication concepts. According to the design, the path is composed of many segments, each of which is composed of (Anchor Point → Gateway). The path discovery process requires the help of all other peers in the middle segment. But the first and last parts are always operated by users who need to access the path[3].

Let's consider, for example, there are two extreme target users named real user and cheaters simulating real user. The cheating controller maintains a trust configuration file for each peer based on the path discovery operation. When the cost of the path segment of the same source and target does not match, it will recalculate the path cost and then the cheater is identified. This process is very simple and it does not always use the controller. Through this process, we can easily capture cheating behavior with a low-cost risk control.

A localized application (plug-in or integrated software) is designed to apply quantitative indicators to the operation of specific objects for data analysis, and ML algorithms is used to learn the user's conventional operation logic and operation content. It is necessary to download and analyze the operation data package with cheating characteristics. When the user who obviously deviates from the normal operation path is close to a certain sub-data in the data package, the application will issue a cheating warning to the user and the server respectively. In this way, the server can target potential users who commit cheating. It also reduces the impact of widely used anti-cheat algorithms on hardware load in some simple operating environments that do not involve economic property security.

4. Discussion and experiment

For example, Final Fantasy XI provides quasi-periodic reports to help to eliminate cheaters. Many terminals have conducted case studies on cheating when discussing the scale of cheating from different angles. Even with full control of the client, the cheater can hide its existence by modifying the operating system,

spoofing anti-cheat software, and more. Therefore, the key step of cheating control is to understand cheating with solutions to the behavior of cheaters.

An Internet shopping application scenario was assumed and the cheating behavior of "swipe order" was stimulated. The computer was used to simulate 10^8 main action frauds. It was found that 99.6% of the scenarios can be directly captured by the local computer through eigenvalue calculation, and the detection efficiency has also increased by 6.29% per month.

5. Conclusions

This paper presents a simple decentralized anti-cheat algorithm and its working mode. This mode discovers the requested path through the participation of peers. When the two peers are inconsistent in path cost, the local controller triggers the implementation of a two-way anti-cheat system. In addition, because the participating peers help during the path discovery process, the load on the server may also decrease. Considering all the characteristics and advantages, it is a feasible solution. Careful research and experiments are still needed to avoid the occurrence of peer-to-peer cheating.

References

1. Bai W. Research on cheat warning model based on user behavior analysis [MSc thesis]. Chengdu: University of Electronic Science and Technology of China; 2018.
2. Ma R. Design and implementation of anti cheating system based on analysis of content and user behavior about social platform [MSc thesis]. Harbin: Harbin Institute of Technology; 2015.
3. Yang Z. Design and implementation of anti-cheat analysis model in O2O scenario [MSc thesis] (in Chinese). Harbin: Harbin Institute of Technology; 2016.